

Solving a Thermal Regenerator Model using Implicit Newton-Krylov Methods*

J. W. Howse G. A. Hansen
D. J. Cagliostro

Computational Science Methods Group, X-CM
P.O. Box 1663, Mail Stop F645
Los Alamos National Laboratory
Los Alamos, NM 87545

K. R. Muske
Department of Chemical Engineering
Villanova University
800 Lancaster Ave.
Villanova, PA 19085

Abstract

In this paper we discuss the use of an implicit Newton-Krylov method to solve a set of partial differential equations representing a physical model of a blast furnace stove. Blast furnace stoves are thermal regenerators used to heat the air injected into the blast furnace, providing the heat to chemically reduce iron oxides to iron. The stoves are modeled using a set of partial differential equations which describe the heat flow in the system. The model is used as part of a predictive controller which minimizes the fuel gas consumption during the heating cycle, while maintaining a high enough output air temperature in the cooling cycle to drive the chemical reaction in the blast furnace. The discrete representation of this model is solved with a preconditioned implicit Newton-Krylov technique. This algorithm uses Newton's method, in which the update to the current solution at each stage is computed by solving a linear system. This linear system is obtained by linearizing the discrete approximation to the PDE's, using a numerical approximation for the Jacobian of the discretized system. This linear system is then solved for the needed update using a preconditioned Krylov subspace projection method.

Nomenclature

T_g :	Gas temperature ($^{\circ}\text{C}$)	A_g :	Area of one tube in the solid (cm^2)
T_s :	Solid temperature ($^{\circ}\text{C}$)	A_s :	Area of the solid surrounding a tube (cm^2)
ρ_g :	Gas density (gm/cm^3)	\mathcal{L}_g :	Perimeter of one tube in the solid (cm)
ρ_s :	Solid density (gm/cm^3)	P_g :	Gas pressure (cal/cm^3)
\mathcal{C}_g :	Gas heat capacity ($\text{cal}/\text{gm}\text{-}^{\circ}\text{K}$)	h :	Heat transfer coefficient ($\text{cal}/\text{cm}^2\text{-sec-}^{\circ}\text{K}$)
\mathcal{C}_s :	Solid heat capacity ($\text{cal}/\text{gm}\text{-}^{\circ}\text{K}$)	\dot{m}_g :	Gas mass flow rate (gm/sec)

*This work was performed under the auspices of the Department of Energy under contract W-7405-ENG-36.

i :	Discrete time index	l :	Linear iteration number
j :	Discrete space index	n :	Total number of nonlinear equations
k :	Nonlinear iteration number		

1 Introduction

This paper presents a set of partial differential equations representing a physical model of a blast furnace stove which are then solved using an implicit Newton-Krylov method. The blast furnace stove is an integral part of the iron making process in the steel industry. These stoves are used to heat air which provides part of the energy required to chemically reduce iron ore to iron metal in the blast furnace. Internally these stoves consist of a combustion chamber and a large mass of refractory brick. The operation of the stoves is functionally divided into two phases. In the first phase (heating cycle), the brick is heated by burning a mixture of combustible exhaust gas from the blast furnace and natural gas in the combustion chamber and allowing this hot exhaust gas to escape through tubes, called flues, in the brick. In the second phase (cooling cycle), air is heated by forcing it through the flues in the heated brick and mixing the resulting hot air with ambient air in the combustion chamber to maintain a constant output temperature. This hot air provides part of the energy needed to drive the desired chemical reactions in the blast furnace. The model is used as part of a predictive control system which reduces the cost of operating these stoves by minimizing the natural gas consumption during the heating cycle, while still maintaining the required output air temperature during the cooling cycle. A diagram of the stove is shown in Figure 1. The chamber on the left in this diagram is where the fuel gas is burned during the heating cycle. The chamber on the right contains the refractory brick, which is in a stack approximately 36 meters high. The insert in Brick Zone #2 shows the shape of one of the bricks, also called checkers. The heating cycle lasts approximately 50 minutes, and the cooling cycle lasts about 30 minutes.

The previous modeling studies of hot blast stoves formed the basis for our model. The first studies by Schofield, Butterfield, and Young (1961) and Hausen and Binder (1962) assumed constant operating conditions, such as gas flow rates and inlet and outlet temperatures, constant physical properties, such as heat capacities and heat transfer coefficients, and assumed that conditions were the same for all sets of heating and cooling cycles. Variable heat capacities and heat transfer coefficients were considered by Butterfield, Schofield, and Young (1963). Varying gas flow rates were considered by Willmott (1968) and Kwakernaak, Strijbos, and Tijssen (1970). In these works, radiation heat transfer from the gas to the solid during the heating cycle were approximated by modifying the convective heat transfer coefficient as discussed by Hausen and Binder (1962). A detailed heat transfer model for blast furnace stoves is presented in this work. We consider variation in all of the physical and operational conditions, both convective and radiative heat transfer mechanisms, flow rate variations during both the heating and cooling cycles, and process changes between sets of heating and cooling cycles.

2 Model Description

The blast furnace stove is modeled by assuming that the gas channels in the checkers can be represented as thick walled tubes in which the gas flows through the center of the tubes thereby heating or cooling the wall material. The outside wall of the tubes are assumed to be perfectly

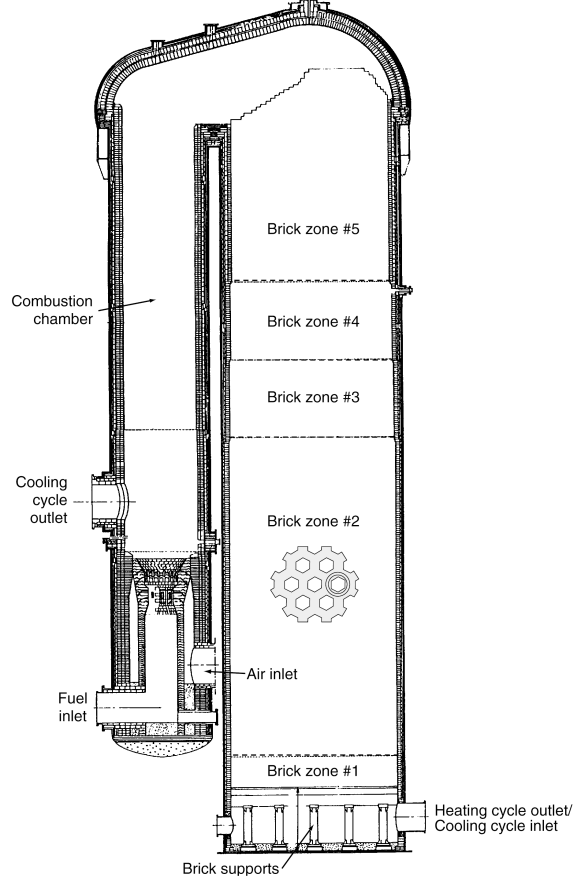


Figure 1: A diagram of a blast furnace stove. The chamber on the left is where combustion of the fuel gas takes place. The chamber on the right contains the refractory brick. The insert in Brick Zone #2 shows the shape of one of the bricks.

insulated. Heat transfer within the stove is modeled with a set of partial differential equations (PDEs) that relate heat flux between the working gas and the storage solid, as presented in Bird, Stewart, and Lightfoot (1960). An energy equation describes the transient heat conduction within the storage brick, and the convective and radiative transport between the working fluid and the storage medium. Specifically, the equations for the energy change in the gas and the solid during the heating cycle are

$$\rho_g c_g \mathcal{A}_g \frac{\partial T_g}{\partial t} = h \mathcal{L}_g (T_s - T_g) - \left(c_g \frac{\partial T_g}{\partial z} + \frac{1}{\rho_g} \frac{\partial P_g}{\partial z} \right) \dot{m}_g(t), \quad (1a)$$

$$\rho_s c_s \mathcal{A}_s \frac{\partial T_s}{\partial t} = h \mathcal{L}_g (T_g - T_s), \quad (1b)$$

$$\rho_g = u(T_g), \quad (1c)$$

$$c_g = v(T_g), \quad (1d)$$

$$c_s = w(T_s), \quad (1e)$$

$$P_g = d(T_g), \quad (1f)$$

$$h = h_c + h_r = c(T_g) + r(T_g, T_s). \quad (1g)$$

In these equations $T_g(t, z)$ is the temperature of the gas, $T_s(t, z)$ is the temperature of the solid, and both vary over time t and space z . The densities of the gas and solid are ρ_g and ρ_s respectively. The heat capacities at constant pressure for the gas and solid are \mathcal{C}_g and \mathcal{C}_s respectively. The quantity \mathcal{L}_g is the perimeter for a single tube in the brick. The quantity \mathcal{A}_g is the area of one tube in the solid, and \mathcal{A}_s is the area of the solid surrounding any one tube. The quantity h is the heat transfer coefficient between the gas and the solid, which consists of a portion h_c due to convection and a portion h_r due to radiation between the gas and the solid. The gas pressure is P_g , and the mass flow rate of gas through the stove is $\dot{m}_g(t)$.

Equation (1a) is the change in energy over time for the gas, while Equation (1b) is the energy change of the solid. The terms in these equations describe the convection of heat between the gas and the solid in the direction perpendicular to the gas flow in the channels, and convection in the gas parallel to the gas flow, other effects are neglected. Note that the only quantity available for controlling the amount of heat in the stove is the mass flow rate $\dot{m}_g(t)$. Equations (1c), (1d), (1e), (1f), and (1g) describe the effect of gas and solid temperatures, T_g and T_s , on the gas density ρ_g , gas heat capacity \mathcal{C}_g , solid heat capacity \mathcal{C}_s , gas pressure P_g , and heat transfer coefficient h . Accounting for temperature variations is necessary because both the gas and solid temperatures vary between 1600°C and 200°C. Equations (1c)–(1g) are described in Appendix A.

Similarly the differential equations for the energy change in the gas and the solid during the cooling phase are

$$\rho_g \mathcal{C}_g \mathcal{A}_g \frac{\partial T_g}{\partial t} = h \mathcal{L}_g (T_s - T_g) - \left(\mathcal{C}_g \frac{\partial T_g}{\partial z} + \frac{1}{\rho_g} \frac{\partial P_g}{\partial z} \right) \dot{m}_g(t), \quad (2a)$$

$$\rho_s \mathcal{C}_s \mathcal{A}_s \frac{\partial T_s}{\partial t} = h \mathcal{L}_g (T_g - T_s), \quad (2b)$$

$$\dot{m}_g(t) = \left(\frac{\int_{T_g^{\text{in}}}^{T_g^{\text{targ}}} \mathcal{C}_g(\tau) d\tau}{\int_{T_g^{\text{in}}}^{T_g^{\text{out}}} \mathcal{C}_g(\tau) d\tau} \right) \dot{m}_g^{\text{in}}, \quad (2c)$$

along with Equations (1c)–(1g). Note that radiative heat transfer is neglected during the cooling phase. The air going into the blast furnace must be maintained at the constant temperature T_g^{targ} . In order to achieve this goal, not all of the air is routed through the stove during the cooling phase. Rather, some of the air is diverted around the stove and is later mixed with the air heated by the stove to maintain the desired outlet temperature. The inlet air temperature T_g^{in} is measured and varies over time. Since the temperature $T_g(t)$ of the air heated by the stove changes over time, the amount of air $\dot{m}_g(t)$ routed through the stove must also change over time. This change in the flow rate through the stove is defined by Equation (2c) and is referred to as the *bypass equation*. The amount of air passing through the stove increases as the cooling phase progresses, due to the cooling of the bricks. The total mass flow rate into the stove–bypass system \dot{m}_g^{in} is measured and varies over time. A diagram depicting the mass and energy flow in the stove is shown in Figure 2. Note that the directions of both heat and mass flow during the cooling cycle are opposite those during the heating cycle. Several things should be noted about this system. First, Equations (1a) and (1b) are identical to Equations (2a) and (2b). These equations are written in terms of flow rate \dot{m}_g rather than gas velocity v_g because flow rate is the quantity that is actually measured and controlled in the stoves. The overall system defined by Equations (1) and (2) contains *both* differential and algebraic equations, hence it is a system of differential algebraic equations (DAE).

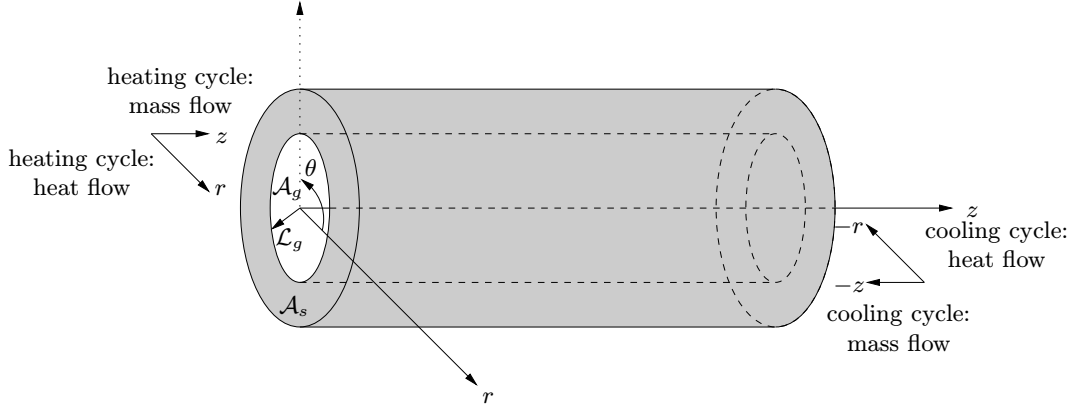


Figure 2: A diagram of the mass and energy flow in the stove for the two phases of its operation. The area of the white circle is \mathcal{A}_g , and the area of the gray annulus is \mathcal{A}_s . The perimeter of the white circle is \mathcal{L}_g . Note that a *cylindrical* coordinate system, centered at the middle of the hole in the brick, is used in this diagram.

The boundary conditions for the heating cycle model in Equation (1) are straightforward to derive. The appropriate conditions are

$$T_g(t, z_{\text{top}}) = T_{\text{comb}}, \quad (3a)$$

$$\left. \frac{\partial T_g}{\partial z} \right|_{z=z_{\text{bot}}} = 0, \quad (3b)$$

$$T_g(t_{\text{heat}}, z) = g(z), \quad (3c)$$

$$T_s(t_{\text{heat}}, z) = g(z), \quad (3d)$$

where z_{top} is the spatial position of the top of the brick stack, z_{bot} is the spatial position of the bottom of the brick stack, and t_{heat} is the time that the heating cycle begins. The quantity T_{comb} is the combustion temperature of the gas burned in the combustion chamber, and the function $g(z)$ is a temperature profile interpolated from temperature measurements of the stove at the end of the previous cooling cycle. Note that the same initial temperature profile is used in both the gas and the solid, because we assume that the temperatures reach local thermal equilibrium during the five minute transition between the heating and cooling cycles. Similarly the boundary conditions for the cooling cycle model in Equation (2) are

$$\left. \frac{\partial T_g}{\partial z} \right|_{z=z_{\text{top}}} = 0, \quad (4a)$$

$$T_g(t, z_{\text{bot}}) = T_{\text{air}}, \quad (4b)$$

$$T_g(t_{\text{cool}}, z) = T_g(t_{\text{comp}}, z), \quad (4c)$$

$$T_s(t_{\text{cool}}, z) = T_s(t_{\text{comp}}, z), \quad (4d)$$

where t_{comp} is the time that the heating cycle ends, and t_{cool} is the time that the cooling cycle begins. The quantity T_{air} is the measured temperature of the input air to the stove during the cooling cycle. The initial temperature profiles in the gas and solid for the cooling cycle are the final temperature profiles computed in the heating cycle. For both heating and cooling cycles the spatial boundary condition is Dirichlet at the inlet side of the brick stack and Neumann at the outlet side.

For computational purposes, Equations (1) and (2) are scaled to make all the quantities dimensionless, and then discretely approximated. The scaling is performed to prevent the choice of units from artificially altering the relative sizes of the terms in Equations (1) and (2). Define the dimensionless scaling variables

$$\begin{aligned}\hat{T}_g &= \frac{T_g - T_i}{T_0}, & \hat{T}_s &= \frac{T_s - T_i}{T_0}, & \hat{P}_g &= \frac{P_g}{P_{g0}}, & \hat{t} &= \frac{t}{\tau_{g0}}, & \hat{z} &= \frac{z}{\tau_{g0} v_{g0}}, & \hat{h} &= \frac{h}{h_0}, \\ \tau_{g0} &= \frac{\rho_{g0} \mathcal{C}_{g0} \mathcal{A}_{g0}}{h_0 \mathcal{L}_{g0}}, & \tau_{s0} &= \frac{\rho_{s0} \mathcal{C}_{s0} \mathcal{A}_{s0}}{h_0 \mathcal{L}_{g0}}, & \hat{\rho}_g &= \frac{\rho_g}{\rho_{g0}}, & \hat{\rho}_s &= \frac{\rho_s}{\rho_{s0}}, & \hat{\mathcal{C}}_g &= \frac{\mathcal{C}_g}{\mathcal{C}_{g0}}, & & (5) \\ P_{g0} &= \rho_{g0} \mathcal{C}_{g0} T_0, & \dot{m}_g(\hat{t}) &= \frac{\dot{m}_g(t)}{\rho_{g0} v_{g0} \mathcal{A}_{g0}}, & \hat{\mathcal{A}}_g &= \frac{\mathcal{A}_g}{\mathcal{A}_{g0}}, & \hat{\mathcal{A}}_s &= \frac{\mathcal{A}_s}{\mathcal{A}_{s0}}, & \hat{\mathcal{L}}_g &= \frac{\mathcal{L}_g}{\mathcal{L}_{g0}}, & \hat{\mathcal{C}}_s &= \frac{\mathcal{C}_s}{\mathcal{C}_{s0}},\end{aligned}$$

where v_{g0} is a reference gas velocity, τ_{g0} is a characteristic time for heat flow in the gas, T_0 is a reference temperature for both the gas and solid, and P_{g0} is a reference gas pressure. Substituting these scaled variables into either Equations (1a) and (1b), or Equations (2a) and (2b) gives the dimensionless partial differential equations

$$\frac{\partial \hat{T}_g}{\partial \hat{t}} = \frac{\hat{h} \hat{\mathcal{L}}_g}{\hat{\rho}_g \hat{\mathcal{C}}_g \hat{\mathcal{A}}_g} (\hat{T}_s - \hat{T}_g) - \left(\frac{1}{\hat{\rho}_g \hat{\mathcal{A}}_g} \frac{\partial \hat{T}_g}{\partial \hat{z}} + \frac{1}{\hat{\rho}_g^2 \hat{\mathcal{C}}_g \hat{\mathcal{A}}_g} \frac{\partial \hat{P}_g}{\partial \hat{z}} \right) \dot{m}_g(\hat{t}) \quad (6a)$$

$$\frac{\partial \hat{T}_s}{\partial \hat{t}} = \frac{\tau_{g0}}{\tau_{s0}} \frac{\hat{h} \hat{\mathcal{L}}_g}{\hat{\rho}_s \hat{\mathcal{C}}_s \hat{\mathcal{A}}_s} (\hat{T}_g - \hat{T}_s) \quad (6b)$$

for both the heating and cooling cycles. In this equation, τ_{s0} is a characteristic time for heat flow in the solid, and $\dot{m}_g(\hat{t})$ is the dimensionless mass flow rate for the gas.

The finite-volume formulation was used to discretize Equation (6) in both space and time. As discussed in Hirsch (1988), the computational domain is divided into some number of non-overlapping control volumes such that only one grid point lies inside each control volume. This set of control volumes must completely cover the original domain. The differential equations are then integrated over each control volume. For discretizing the system, denote the states of the system by the vector $\mathbf{x}^\dagger = [T_g \ T_s]$. These integrals are evaluated by approximating the variation of \mathbf{x} between each grid point using piecewise profiles. This procedure results in a set of discrete equations containing values of \mathbf{x} for each grid point. Intuitively these discrete equations define a conservation principle for \mathbf{x} over the finite volume of each cell, just as the original differential equations express it for an infinitesimal volume. The algebraic equations are assumed to be constant over each control volume, so they can be removed from the volume integrals. As an example consider discretizing an equation of the form $\mathcal{C}_g \frac{\partial T_g}{\partial z} \dot{m}_g(t) = 0$. Applying Gauss' theorem to the volume integral gives $\int_V \mathcal{C}_g \dot{m}_g(t) \frac{\partial T_g}{\partial z} dV = \mathcal{C}_g \int_V \nabla \cdot (\dot{m}_g T_g) dV = \mathcal{C}_g \oint_S (\dot{m}_g T_g) \cdot \hat{n} dS = 0$. In our problem, the dimensions of each control volume are identical, hence the discretized form of this equation for the j th space point at the i th time point is $\mathcal{C}_g(i, j) \dot{m}_g(i) \frac{T_g(i, j+1) - T_g(i, j)}{\Delta z} = 0$. In this particular example the result is identical to that obtained with the finite-difference method, but in general this is not necessarily the case. Since this is a convection problem, the upwind-difference scheme is used to discretize the term $\frac{\partial \hat{T}_g}{\partial \hat{z}}$. This approach approximates the derivative of the temperature at a node by taking the difference between the temperature at that node and the temperature at the preceding node relative to the direction of the flow. Therefore if the direction of flow changes, then the node defined as preceding also changes. This occurs in our problem since the direction of gas flow through the stove reverses between the heating and cooling cycles.

Using this approach the discretized form of the scaled differential equations in Equation (6) is

$$\begin{aligned} \frac{\hat{T}_g(i, j) - \hat{T}_g(i-1, j)}{\Delta t} &= \frac{\hat{h}(i, j) \hat{\mathcal{L}}_g(i, j)}{\hat{\rho}_g(i, j) \hat{\mathcal{C}}_g(i, j) \hat{\mathcal{A}}_g(i, j)} (\hat{T}_s(i, j) - \hat{T}_g(i, j)) \\ &\quad - \frac{1}{\hat{\rho}_g(i, j) \hat{\mathcal{A}}_g(i, j)} \left(\frac{\hat{T}_g(i, j) - \hat{T}_g(i, j-1)}{\Delta z} \right) \dot{m}_g(i) \\ &\quad - \frac{1}{\hat{\rho}_g^2(i, j) \hat{\mathcal{C}}_g(i, j) \hat{\mathcal{A}}_g(i, j)} \left(\frac{\hat{P}_g(i, j+1) - \hat{P}_g(i, j-1)}{2 \Delta z} \right) \dot{m}_g(i), \end{aligned} \quad (7a)$$

$$\begin{aligned} \frac{\hat{T}_g(i, j) - \hat{T}_g(i-1, j)}{\Delta t} &= \frac{\hat{h}(i, j) \hat{\mathcal{L}}_g(i, j)}{\hat{\rho}_g(i, j) \hat{\mathcal{C}}_g(i, j) \hat{\mathcal{A}}_g(i, j)} (\hat{T}_s(i, j) - \hat{T}_g(i, j)) \\ &\quad - \frac{1}{\hat{\rho}_g(i, j) \hat{\mathcal{A}}_g(i, j)} \left(\frac{\hat{T}_g(i, j+1) - \hat{T}_g(i, j)}{\Delta z} \right) \dot{m}_g(i) \\ &\quad - \frac{1}{\hat{\rho}_g^2(i, j) \hat{\mathcal{C}}_g(i, j) \hat{\mathcal{A}}_g(i, j)} \left(\frac{\hat{P}_g(i, j+1) - \hat{P}_g(i, j-1)}{2 \Delta z} \right) \dot{m}_g(i), \end{aligned} \quad (7b)$$

$$\frac{\hat{T}_s(i, j) - \hat{T}_s(i-1, j)}{\Delta t} = \frac{\tau_{g0}}{\tau_{s0}} \frac{\hat{h}(i, j) \hat{\mathcal{L}}_g(i, j)}{\hat{\rho}_s(i, j) \hat{\mathcal{C}}_s(i, j) \hat{\mathcal{A}}_s(i, j)} (\hat{T}_g(i, j) - \hat{T}_s(i, j)), \quad (7c)$$

where j is the space node index and i is the time node index. Equation (7a) is the discrete energy equation for the gas during the heating cycle, Equation (7b) is the gas energy variation for the cooling cycle, and Equation (7c) is the solid energy change for both cycles. A time grid was used because the mass flow rate \dot{m}_g during the cooling cycle varies over time due to the temperature changes, as shown in Equation (2c). For simplicity we wanted to use the same solution approach for both the heating and cooling cycles, hence a time grid is also used during the heating cycle. The conditions at the initial time become conditions along a boundary of the time dimension of our two dimensional grid. This treatment of time requires the temporal derivatives to be incorporated into the Jacobian of the discrete system in the same manner that spatial derivatives are normally incorporated.

3 Solution Technique

We chose an implicit Newton-Krylov technique to solve the discrete representation in Equation (7) because the solution technique must be robust for systems having disparate eigenvalues in the linear approximation, and it must provide rapid convergence without using tuning parameters. The disparity in eigenvalues is created by the different time scales for convection in the gas, and conduction in the brick. Rapid convergence is required in order to allow the controller to compute the optimal mass flow rate for the fuel gas during the heating cycle during the short time in between the end of a cooling cycle and the beginning of a heating cycle. Lastly, a parameter-free method allows the use of the technique by operating personnel with limited experience in non-linear solution techniques. The algorithm is an inexact version of Newton's method, where the update to the current solution at each stage is computed by approximately solving a linear system. This linear system results from linearizing the discrete approximation to the PDE's, using a numerical approximation for the Jacobian of the discretized system. The resulting linear system is solved iteratively for the needed update using a preconditioned Krylov subspace projection method. Various methods of this type are discussed by Kelley (1995).

As indicated above, our solution technique breaks down naturally into two parts. The first part consists of searching for a nonlinear update to the current solution. Conceptually, Equation (7) can be rewritten as the vector equation $\mathbf{f}(T_g(i, j), T_s(i, j)) = \mathbf{0}$. An approximate solution to this differential algebraic system is given by a set of states $T_g(i, j)$, and $T_s(i, j)$ which make the value of $\mathbf{f}(\cdot)$ close to zero for each space point j and time point i . Intuitively, this is a root finding problem in which the roots are *functions* of the distance j and the time i . The function $\mathbf{f}(\cdot)$ is called the *nonlinear residual*. Collectively the states are denoted by the vector $\mathbf{x}^\dagger = [T_g(i, j) \ T_s(i, j)]$. The root finding problem is to find the state \mathbf{x} which minimizes the nonlinear residual $\mathbf{f}(\mathbf{x})$. One way to solve this problem is to compute the second order Taylor series expansion of $\mathbf{f}(\mathbf{x})$ about the point \mathbf{x}

$$\mathbf{f}_i(\mathbf{x} + \delta\mathbf{x}) = \mathbf{f}_i(\mathbf{x}) + \sum_{m=1}^n \frac{\partial \mathbf{f}_m}{\partial x_m} \delta x_m + O(\delta\mathbf{x}^2), \quad (8)$$

where $n = n_v n_s n_t$ is the product of the number of state variables (*i.e.*, $n_v = 2$ in our case), the number of grid nodes in space n_s , and the number of grid nodes in time n_t . Neglecting terms of order $\delta\mathbf{x}^2$ and higher and setting $\mathbf{f}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{0}$, we obtain a set of linear equations for the corrections $\delta\mathbf{x}$ that move each residual toward zero simultaneously. For the k th iteration of the algorithm, the vector form of these equations is

$$\mathbf{J}_f(\mathbf{x}_k) \delta\mathbf{x}_k = -\mathbf{f}(\mathbf{x}_k), \quad (9)$$

where $\mathbf{J}_f(\mathbf{x}_k)$ is the Jacobian matrix of the discrete system $\frac{\partial \mathbf{f}_k}{\partial \mathbf{x}_k}$. The corrections are added to the solution vector giving the update rule

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \delta\mathbf{x}_k, \quad (10)$$

where $\alpha_k \in (0, 1]$ is a weighting factor to keep the algorithm from overshooting the solution. This algorithm for root solving is commonly known as the *Newton-Raphson method* or simply *Newton's method*, as detailed in Fletcher (1987). The algebraic equations of state are recomputed using the new temperature estimate \mathbf{x}_{k+1} after each Newton iteration. The Newton iterations are stopped when the criteria

$$\|\mathbf{f}(\mathbf{x}_k)\|_2 < \tau_r \|\mathbf{f}(\mathbf{x}_0)\|_2 + \tau_a \quad (11)$$

is satisfied. In this equation $\tau_r \in (0, 1)$ is the relative error tolerance, $\tau_a \in (0, 1)$ is the absolute error tolerance, and $\|\cdot\|_2$ is the Euclidean norm. Intuitively this criteria means that the Newton iterations are stopped when either the current residual $\|\mathbf{f}(\mathbf{x}_k)\|_2$ becomes less than τ_r of its initial value $\|\mathbf{f}(\mathbf{x}_0)\|_2$, or when it becomes less than τ_a .

The second part of the algorithm consists of finding the solution for the linear system in Equation (9). This equation is of the general form $\mathbf{A}\mathbf{y} = \mathbf{b}$, where \mathbf{A} is an $(n \times n)$ matrix. The method that we use is a conjugate-gradient-like polynomial-based iterative scheme. The general solution update is

$$\mathbf{y}_l = \mathbf{y}_0 + (\gamma_{l0} \mathbf{r}_0 + \gamma_{l1} \mathbf{A} \mathbf{r}_0 + \gamma_{l2} \mathbf{A}^2 \mathbf{r}_0 + \cdots + \gamma_{l(l-1)} \mathbf{A}^{l-1} \mathbf{r}_0), \quad (12)$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A} \mathbf{y}_0$, \mathbf{r}_l is the linear residual at step l , and \mathbf{y}_0 is the initial guess for the solution of the linear system. This means that the solution \mathbf{y}_l at step l is the initial solution \mathbf{y}_0 plus a linear combination of vectors in the set $\{\mathbf{r}_0, \mathbf{A} \mathbf{r}_0, \mathbf{A}^2 \mathbf{r}_0, \dots, \mathbf{A}^{l-1} \mathbf{r}_0\}$. The space spanned by this set of vectors is the *Krylov subspace*, which is denoted by $\mathcal{K}_l(\mathbf{r}_0, \mathbf{A})$. Since new solution

approximations are computed by projecting the linear residual \mathbf{r}_l onto a Krylov subspace, these algorithms are collectively known as Krylov subspace projection methods, as discussed by Saad (1996).

Equation (12) can be written in the simpler form $\mathbf{y}_l = \mathbf{y}_0 + \sum_{m=0}^l \gamma_{lm} \mathbf{p}_m$. The manner in which \mathbf{p}_l is computed defines a particular Krylov subspace method. In general, two criteria can be used to compute the \mathbf{p}_l vectors. The first criterion is to pick \mathbf{p}_l to minimize some norm of the current linear residual \mathbf{r}_l . The second criterion is to choose \mathbf{p}_l so that the the current linear residual \mathbf{r}_l is orthogonal to some set of vectors \mathcal{L}_l , where \mathcal{L}_l may be different from \mathcal{K}_l . Mathematically, these two criteria are

$$\min_{\mathbf{p}_l \in \mathcal{K}_l} \|\mathbf{r}_l\|_{\mathcal{N}} = \min_{\mathbf{p}_l \in \mathcal{K}_l} \left\| \mathbf{r}_0 - \mathbf{A} \sum_{m=0}^l \gamma_{lm} \mathbf{p}_m \right\|_{\mathcal{N}}, \quad (13)$$

$$\mathbf{r}_l = \left(\mathbf{r}_0 - \mathbf{A} \sum_{m=0}^l \gamma_{lm} \mathbf{p}_m \right) \perp \mathcal{L}_l, \quad (14)$$

where $\|\cdot\|_{\mathcal{N}}$ represents an arbitrary norm. By satisfying the first criterion, the algorithm is guaranteed to converge to a solution which minimizes some measure of the error between the exact and approximate solutions. By satisfying the second criterion, the algorithm is guaranteed to converge in a finite number of iterations. The conjugate gradient algorithm is derived assuming that \mathbf{A} is symmetric positive definite, in which case both of these criteria can be satisfied simultaneously with $\mathcal{L}_l \equiv \mathcal{K}_l$.

In most cases the Jacobian is *not* symmetric positive definite, hence both of the above criteria can not be satisfied simultaneously. There are numerous algorithms based on different implementations of one of these two criteria. The technique that we use is the Generalized Minimal Residual (GMRES) algorithm developed by Saad and Schultz (1986). This algorithm has three distinguishing features. First, it is guaranteed to minimize the 2-norm of the linear residual $\|\mathbf{r}_l\|_2 = \left\| \mathbf{r}_0 - \mathbf{A} \sum_{m=0}^l \gamma_{lm} \mathbf{p}_m \right\|_2 = \|\mathbf{b} - \mathbf{A} \mathbf{y}_l\|_2$. Second, the search directions \mathbf{p}_l are \mathbf{I} -orthonormal, meaning that $\mathbf{p}_m^\dagger \mathbf{p}_n = 0$ for all $m \neq n$, and $\|\mathbf{p}_m\|_2 = 1$. Third, the linear residual at any iteration is \mathbf{A} -orthogonal to all previous search directions, meaning $\mathbf{r}_m^\dagger \mathbf{A} \mathbf{p}_n = 0$ for all $m > n$. Another way to state the last condition is that the linear residual \mathbf{r}_l is orthogonal to the Krylov subspace $\mathcal{L}_l = \mathbf{A} \mathcal{K}_l(\mathbf{r}_0, \mathbf{A})$. The stopping criteria for the GMRES iterations is

$$\|\mathbf{r}_l\|_2 < \epsilon_r \|\mathbf{r}_0\|_2 + \epsilon_a, \quad (15)$$

where $\epsilon_r \in (0, 1)$ is the relative error tolerance and $\epsilon_a \in (0, 1)$ is the absolute error tolerance. The explanation of this criteria is the same as that for the Newton stopping criteria.

The speed of convergence for finding the solution \mathbf{y} of the linear system $\mathbf{A} \mathbf{y} = \mathbf{b}$ depends on the ratio of the maximum to the minimum eigenvalues of the matrix \mathbf{A} . If this ratio is large, then \mathbf{A} is said to be poorly conditioned. In many practical cases \mathbf{A} is so poorly conditioned that Krylov methods, such as GMRES, do not converge at all. Preconditioning makes a linear system easier to solve by improving its condition number. Preconditioning is accomplished by multiplying both sides of the linear system by a matrix \mathbf{P} which resembles \mathbf{A}^{-1} in some sense. The matrix \mathbf{P} can be multiplied on either the left or right hand sides of the original system, giving rise to one of two new systems

$$\mathbf{P} \mathbf{A} \mathbf{y} = \mathbf{P} \mathbf{b}, \quad (16a)$$

$$\mathbf{A} \mathbf{P} \mathbf{u} = \mathbf{b}, \quad \mathbf{u} = \mathbf{P}^{-1} \mathbf{y}. \quad (16b)$$

Equation (16a) is the new system under left preconditioning, and Equation (16b) is the new system under right preconditioning. In our application a right preconditioner is applied to Equation (9) prior to solving for the linear correction $\delta \mathbf{x}_k$ using GMRES.

The preconditioner used in this application is an incomplete LU factorization of the Jacobian matrix $\mathbf{J}_f(\mathbf{x}_k)$ with threshold dropping and diagonal compensation, called MILUT by Saad (1996). Incomplete LU factorization consists of performing Gaussian elimination on the matrix \mathbf{A} and dropping some elements from the intermediate matrix LU_m at each step m of elimination. This procedure guarantees that the matrix obtained at the final step of this incomplete factorization LU_n is sparse, which would generally *not* be the case for the matrix obtained by Gaussian elimination. At the m th elimination step, the element $LU_m(u, v)$ is dropped if $|LU_m(u, v)| < \eta \frac{\|LU_m(u, \cdot)\|_\infty}{|LU_m(u, u)|}$, where $LU_m(u, \cdot)$ is the u th row of the matrix LU_m , $LU_m(u, u)$ is the diagonal element of the u th row, $\eta \in (0, \infty)$ is the drop tolerance parameter, and $\|\cdot\|_\infty$ is the infinity norm. The diagonal entries $LU_m(u, u)$ at each iteration are compensated for the dropped terms by subtracting a weighted sum of the dropped terms from the diagonal entry $LU_m(u, u) = LU_m^*(u, u) - \mu \sum_{q=\text{drop}} LU_m(u, q)$, where $LU_m^*(u, u)$ is the diagonal entry computed by elimination, and $\mu \in [0, 1]$ is the compensation parameter. The matrix obtained after the last step of incomplete elimination LU_n is used as the preconditioner matrix \mathbf{P} , so $\mathbf{P} = LU_n$.

This solution technique has numerous positive features. It has been proven in Dembo, Eisenstat, and Steihaug (1982) that the upper bound on the convergence rate of inexact Newton's method is superlinear. Likewise it was shown in van der Vorst and Vuk (1993) that the convergence bound for GMRES is also superlinear. Since the algorithm is implicit we can follow any time or space scale in the problem, rather than being forced to follow the fastest time scale or smallest space scale, as in explicit methods. This algorithm directly minimizes both the absolute and relative error of the solution. Because this method is based on root finding, the resulting solution is one for which $\mathbf{f}(\mathbf{x}_k) \approx \mathbf{0}$, and $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathcal{N}} \approx 0$ for some iteration k . Also this algorithm has modest memory requirements, and has very few, if any, parameters.

This solution algorithm also has some negative features. In practice Newton's method often diverges unless it is started fairly close to a root. Furthermore, for roots with order greater than one, the upper bound on the convergence rate is linear. In this application, although both of these difficulties are still possible, there is one feature of the problem which simplifies matters. For a well-posed system of differential equations, there is a unique real-valued solution which depends continuously on the initial and boundary conditions. *Assuming* that this is also true for the discretized system, there is only *one real root* for $\mathbf{f}(\mathbf{x}_k) = \mathbf{0}$. The fact that there is only one real root may simplify the task of computing it. Newton's method can be made more robust to the initial guess by adjusting the size of the Newton step taken in each iteration k using the parameter α_k in Equation (10). A number of line search methods for accomplishing this are discussed in Dennis and Schnabel (1983). Another potential difficulty is that GMRES is not guaranteed to converge in a finite number of iterations. This difficulty is dealt with by preconditioning the linear system in Equation (9). The goal of preconditioning is to make this equation much easier to solve without expending much computational effort constructing the preconditioner.

4 Simulation Results

First we will discuss the agreement between our model and measurements from the stoves. One measure of the state of the stove during the heating and cooling cycles is the temperature of the bricks in the stove. Three temperature measurements are made within the brick stack in the stove. The first is at the top of the brick stack (dome), the second is 11 meters down from the top of the stack (interface), and the third is at the bottom of the stack (grid). These temperature measurements are recorded at 15 second intervals throughout each stove cycle. Another measure of the state of the stove for which data is available is the amount of air sent through the stove during the cooling cycle. This is expressed as the mass flow rate through the stove as a function of time during the cooling cycle, and it is measured every 15 seconds. Using the model, it is possible to compute a temperature at each of the measurement positions over a heating and cooling cycle and a mass flow rate through the stove during the cooling cycle. Four plots comparing the actual and simulated time evolution of these four quantities are shown in Figure 3.

In each of these graphs the solid lines are the measured quantities and the dashed lines are the simulated quantities. All simulated temperatures shown are solid temperatures. Figure 3(a) shows the dome temperature versus time, Figure 3(b) plots time evolution of the interface temperature, Figure 3(c) illustrates the grid temperature over time, and Figure 3(d) graphs the mass flow rate through the stove during a cooling cycle. Based on these plots there appears to be good agreement between our model and the measured data, except for the dome temperature. As shown by Figure 3(a), the measured dome temperature does not change very much over time. We believe that the dome temperature sensors are unreliable and inaccurate and consequently these readings are poor measurements of the brick temperature at the top of the stack. The disparity between the measured and computed mass flow rate through the stove in Figure 3(d) is due to neglecting the heat stored in the brick lining of the combustion chamber in our simulation. These plots illustrate that the system does not reach thermal equilibrium during either the heating or cooling cycle, since the time derivatives of the temperature measurements are clearly not zero during these cycles. Note that there are only three temperature measurements being made down the entire length of the stove. One problem in assessing the validity of the model is this lack of measurements from the actual system. The flat spots in the simulated temperature values are due to the presence of two cycles of stove operation that are unmodeled. The first unmodeled cycle occurs between the end of a heating cycle and the beginning of a cooling cycle. In this cycle, the pressure inside the stove is increased from one atmosphere to five atmospheres over a period of five minutes. The second unmodeled cycle occurs between the end of a cooling cycle and the beginning of a heating cycle. In this cycle, the pressure inside the stove is decreased from five atmospheres to one atmosphere over a period of five minutes. Our studies indicate that the initial and final solid temperatures are roughly equal for both of these cycles, therefore we did not attempt to model them.

Next we will discuss the convergence properties of our Newton-Krylov implementation with respect to both the number of Newton iterations to approximately solve $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ and the number of linear iterations to approximately solve Equation (9). As discussed in Luenberger (1984), the rate of convergence of a sequence $\{\mathbf{u}_n\}_{n=0}^{\infty}$ which converges to a limit \mathbf{u}^* can be assessed by computing $\beta = \lim_{n \rightarrow \infty} \frac{\|\mathbf{u}_{n+1} - \mathbf{u}^*\|}{\|\mathbf{u}_n - \mathbf{u}^*\|^p}$ where p is a positive integer. The *order of convergence* is the largest number p for which $0 \leq \beta < \infty$. If $p = 1$ and $0 < \beta < 1$, then the convergence rate is said to be linear. If $p = 1$ and $\beta = 0$, then the convergence rate is superlinear. If $p = 2$, then the convergence rate is quadratic. For example, given a real number a such that $0 < a < 1$, the

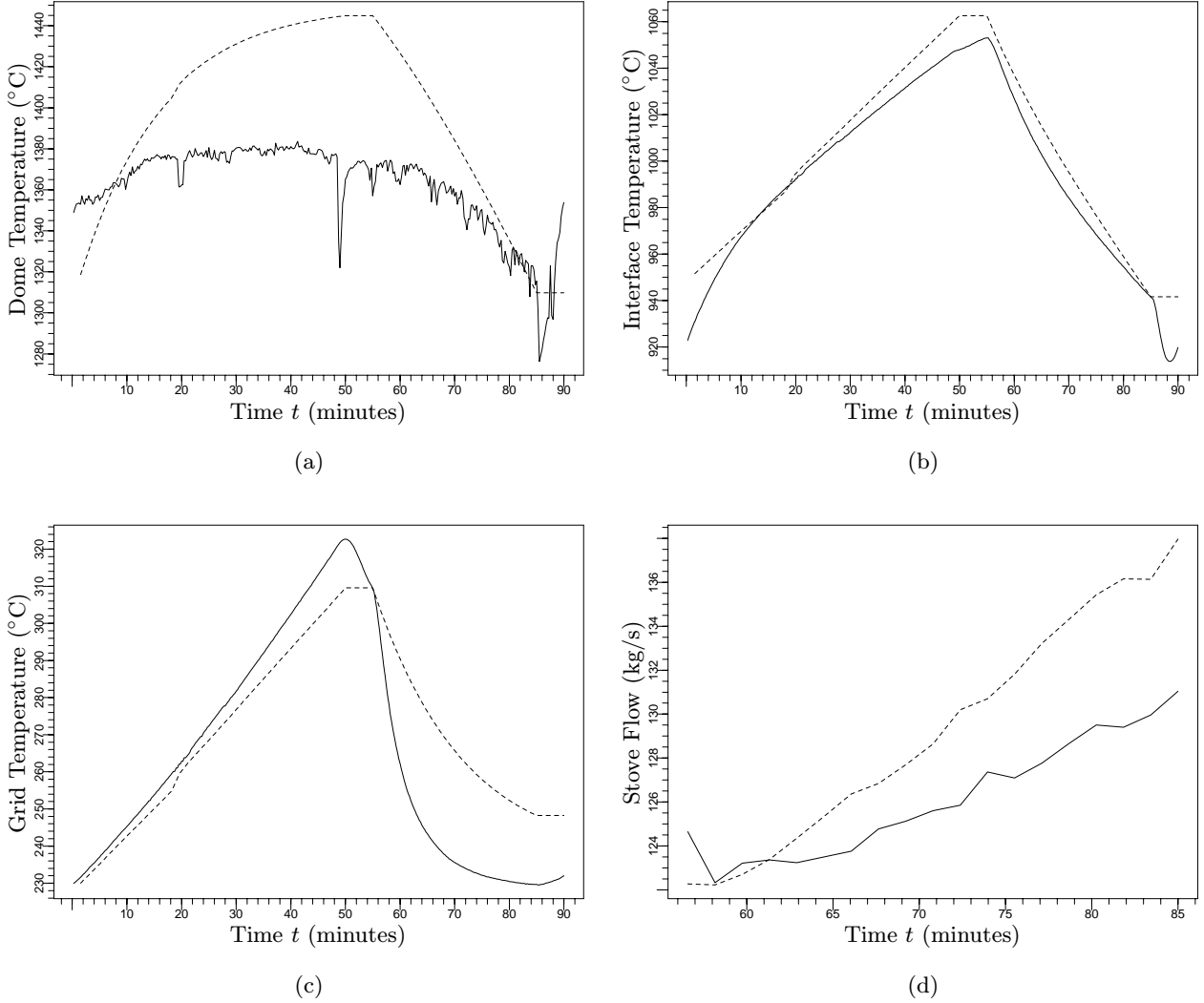


Figure 3: (a) The dome temperature in $^{\circ}\text{C}$ versus time in minutes for both a heating and cooling cycle. The solid line is the measured data and the dashed line is the simulated data.

(b) The interface temperature versus time.

(c) The grid temperature versus time.

(d) The mass flow rate through the stove in kg/s versus time in minutes for a cooling cycle.

sequence $u_n = a^n$ converges linearly, the sequence $u_n = a^{n^2}$ converges superlinearly, and $u_n = a^{2^n}$ converges quadratically. The convergence properties of the Newton-Krylov algorithm for this application are illustrated in Figure 4. Figures 4(a) and 4(b) plot the nonlinear residual $\|\mathbf{f}(\mathbf{x}_k)\|_2$ versus the nonlinear iteration k during a heating cycle and a cooling cycle respectively. These two figures show the convergence of Newton's method while solving for the final solution \mathbf{x}_{fin} using Equation (10). Similarly, Figures 4(c) and 4(d) plot the linear residual $\|\mathbf{f}(\mathbf{x}_k) + \mathbf{J}_{\mathbf{f}}(\mathbf{x}_k) \delta \mathbf{x}_l\|_2$ versus the total number of linear iterations l during a heating cycle and a cooling cycle respectively. These two figures show the convergence of MILUT preconditioned GMRES while solving for $\delta \mathbf{x}_k$ with Equation (12) during each nonlinear iteration of each cycle. Note that Figures 4(c) and 4(d) show all the linear iterations taken during a heating cycle and a cooling cycle respectively. The points connected by the dashed lines in these two figures are the nonlinear residual values

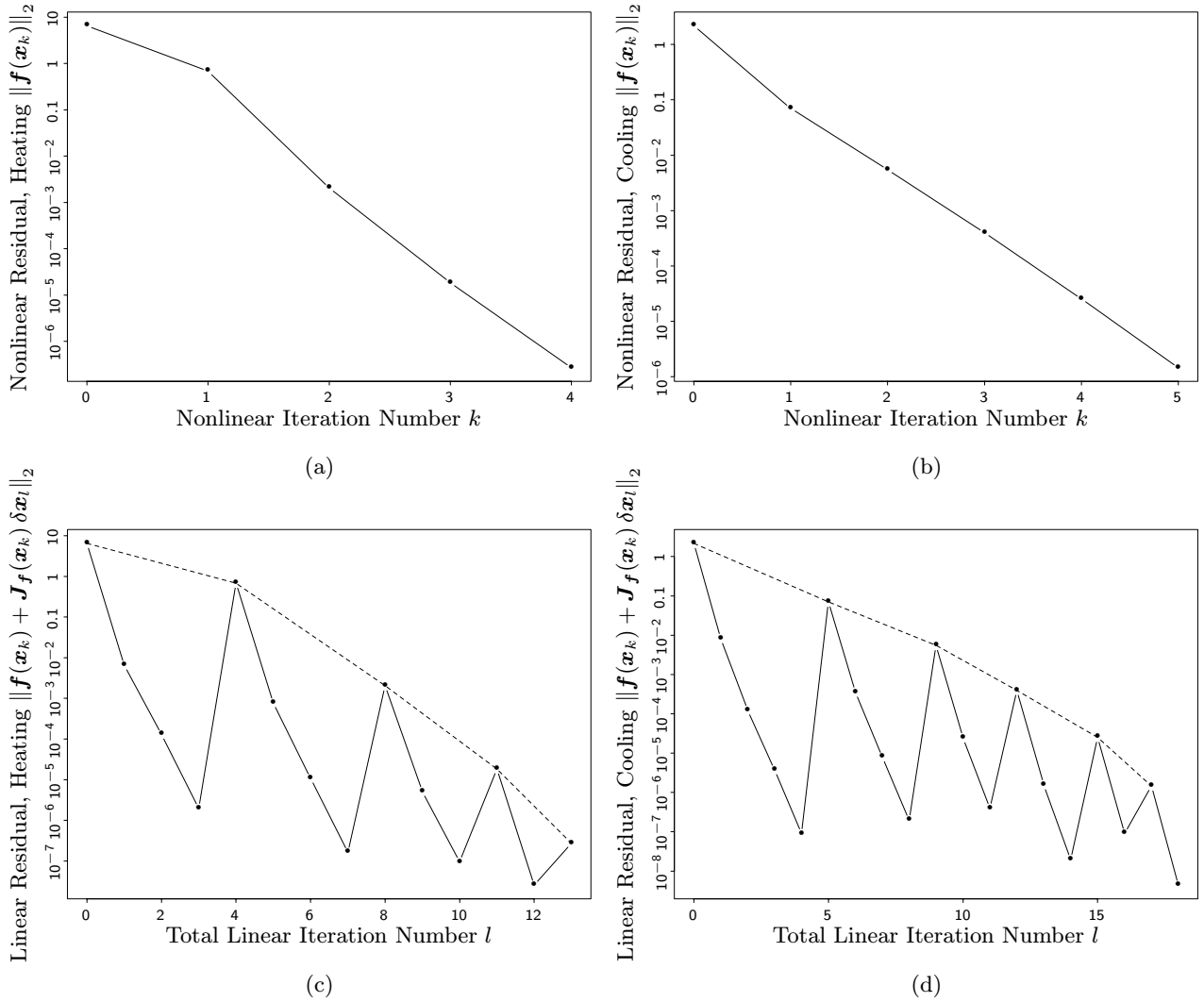


Figure 4: (a) The convergence of the residual $\|f(x_k)\|_2$ for Newton's method while solving for x_{fin} using Equation (10) during a heating cycle.

(b) The residual convergence for Newton's method during a cooling cycle.

(c) The convergence of the residual $\|f(x_k) + J_f(x_k) \delta x_l\|_2$ for MILUT preconditioned GMRES while solving for δx_k in Equation (9) during a heating cycle.

(d) The residual convergence for MILUT preconditioned GMRES during a cooling cycle.

shown in Figures 4(a) and 4(b). This means that the number of linear iterations for each nonlinear iteration can be deduced by counting the number of points between adjacent points on the dashed line. Notice that the vertical axes of all these plots are logarithmic. Intuitively, on a semi-logarithmic plot, a sequence which converges linearly will appear as a straight line. A sequence converging faster than linearly will have a negative curvature (*i.e.*, curving downward) and one converging slower than linearly will have positive curvature (*i.e.*, curving upward). Therefore Figures 4(c) and 4(d) indicate that MILUT preconditioned GMRES is converging superlinearly. Figures 4(a) and 4(b) indicate that Newton's method is converging linearly. We believe that the large number of algebraic equations in our model is limiting the nonlinear convergence rate. Another way to assess convergence rate is by computing the number of iterations required to

reduce the the residual by a factor of 10. This quantity is computed by evaluating the expression $\lambda = \frac{k_f - k_i}{\log(\|\mathbf{f}(\mathbf{x}_{k_i})\|_2) - \log(\|\mathbf{f}(\mathbf{x}_{k_f})\|_2)}$ where k_i is the initial and k_f is the final nonlinear iteration number. For the heating cycle shown in Figure 4(a), $\lambda_{\text{heat}} = 0.542$ nonlinear iterations are required to reduce the residual $\|\mathbf{f}(\mathbf{x}_k)\|_2$ by a factor of 10, and for the cooling cycle in Figure 4(b) $\lambda_{\text{cool}} = 0.809$ iterations are needed. So in spite of the fact that the nonlinear convergence rate is linear, the rate of decrease for the residual is acceptably fast.

We also investigated convergence with respect to the number of time and space nodes for the discrete approximation in Equation (7). Intuitively, as the the number of nodes is increased, the cell volume in the discretization is decreased, and at some point the changes in the simulation results should become very small. One appropriate metric for evaluating the simulation results is the error between the energy change in the gas and the energy change in the solid for a heating or cooling cycle. The energy changes in the gas and solid for the discrete system, and the relative error between these two quantities, are given by

$$\Delta E_g = \sum_{j=1}^{n_t} \dot{m}_g(j) \left(\int_{T_g^{\text{in}}}^{T_g^{\text{out}}} C_g(T_g) dT_g \right) \Delta t, \quad (17a)$$

$$\Delta E_s = \sum_{i=1}^{n_s} \rho_s(i) \mathcal{A}_s \left(\int_{T_s^{\text{init}}}^{T_s^{\text{fin}}} C_s(T_s) dT_s \right) \Delta z, \quad (17b)$$

$$\Delta E_{\text{error}} = 100 \frac{\Delta E_g - \Delta E_s}{\Delta E_s}. \quad (17c)$$

In this equation, T_g^{in} and T_g^{out} are the inlet and outlet gas temperatures respectively, T_s^{init} and T_s^{fin} are the initial and final solid temperatures, n_t is the number of time nodes, n_s is the number of space nodes, Δt is the size of one time step, and Δz is the size of one space step. As the number of time nodes goes to infinity $n_t \rightarrow \infty$, the sum in Equation (17a) becomes an integral over time t . Likewise, as the number of space nodes goes to infinity $n_s \rightarrow \infty$, the sum in Equation (17b) becomes an integral over space z .

We performed a grid convergence study by computing the relative energy error ΔE_{error} as a function of both the number of time nodes n_t and the number of space nodes n_s . In this study the number of time nodes for the cooling cycle was one half the number of time nodes during the heating cycle because the cooling cycle is roughly half as long as the heating cycle. The results of these computations are plotted in Figure 5. Figures 5(a) and 5(b) plot the relative energy error ΔE_{error} versus the number of time nodes n_t for various space node values for the heating and cooling cycles respectively. In these two graphs, each set of similar points connected by a dotted line corresponds to a different number of space nodes. Similarly, Figures 5(c) and 5(d) plot the relative energy error ΔE_{error} versus the number of space nodes n_s for various time node values for the heating and cooling cycles respectively. Likewise, for these two graphs each set of connected similar points represents a different number of time nodes. It is clear from these plots that the relative energy error ΔE_{error} approaches zero for a sufficiently large number of time and space nodes during both the heating and cooling cycles. There are errors associated with both the discrete approximation of the differential equations in Equation (7) and the discrete approximation of the integrals in Equation (17). The effect of these errors is most clearly seen in Figures 5(b) and 5(d) where ΔE_{error} is positive for a small number of nodes and becomes negative for a large number of nodes.

To minimize the computation time while still maintaining sufficient accuracy, we chose the

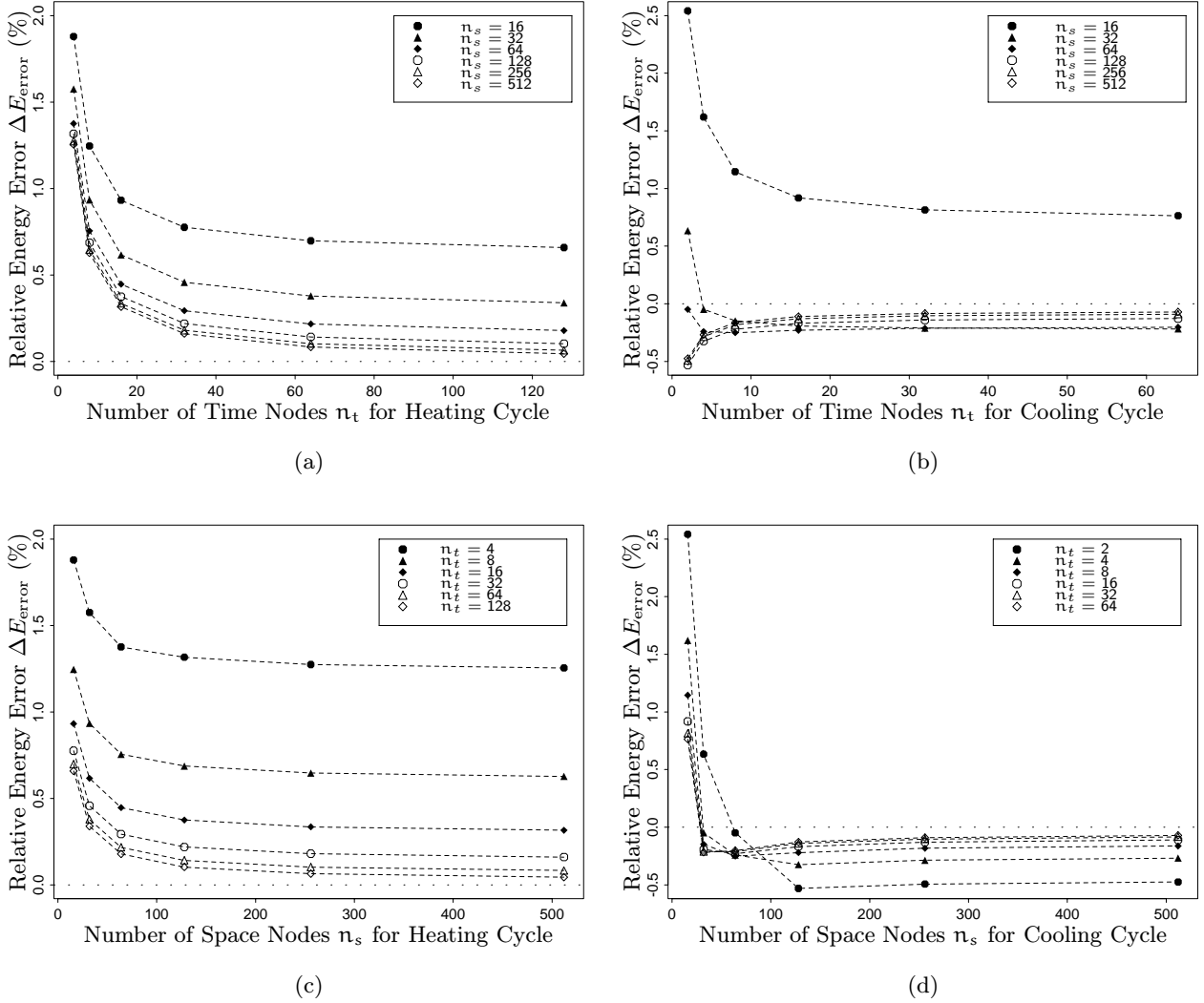


Figure 5: (a) The relative energy error ΔE_{error} versus the number of time nodes n_t for the heating cycle for various numbers of space nodes n_s .

(b) The energy error versus the number of time nodes for the cooling cycle.

(c) The relative energy error ΔE_{error} versus the number of space nodes n_s for the heating cycle for various numbers of time nodes n_t .

(d) The energy error versus the number of space nodes for the cooling cycle.

coarsest grid for which the solution became invariant to the number of grid nodes based on Figure 5. Specifically we used 100 space nodes over the 36 meter length of the stove, 30 time nodes over the 50 minute heating cycle, and 20 time nodes over the 30 minute cooling cycle. With these choices for the grid spacing the relative energy balance error is on the order of 0.25%. This error can be reduced by using additional grid points, as shown in Figure 5, at the expense of greater computation time. For the purposes of this model, a relative energy balance error on the order of 0.25% is acceptable. These grid spacing choices mean that there are $n = 6000$ nonlinear equations $\mathbf{f}(\cdot)$ of the forms given by Equations (7a) and (7c) during the heating cycle, and $n = 4000$ nonlinear equations $\mathbf{f}(\cdot)$ as specified in Equations (7b) and (7c) during the cooling cycle. It is clear from Equations (7a) and (7b) that the portion of the Jacobian $\mathbf{J}_{\mathbf{f}}(\cdot)$ associated

with the discrete gas energy equations has four bands. One band is the diagonal, and the other three are off-diagonal. The off-diagonal bands are associated with the temporal derivatives, the spatial derivatives, and the coupling between the gas and the solid, with one band for each of these terms. Likewise from Equation (7c), the portion of the Jacobian $\mathbf{J}_f(\cdot)$ for the discrete solid energy equations has only three bands, because there are no spatial derivatives in this equation.

5 Conclusion

In this paper we have presented a physical model of a blast furnace stove which was solved with an implicit Newton-Krylov algorithm. Our heat transfer model allows for variation of both physical and operational parameters and utilizes transient dynamics. The model is used as part of an adaptive predictive controller which minimizes the natural gas consumption during the heating cycle, while still maintaining the necessary air temperature during the cooling cycle. This control system has been running for over six months and company audits indicate that it has reduced natural gas usage by 5%, as detailed in Muske *et al.* (1999). The Newton-Krylov technique was selected for several reasons. It is robust for solving systems having components which evolve at very different time scales. In this application, this problem is particularly acute during the cooling cycle wherein the time scale of the bypass computation in Equation (2c) is much faster than the time scale of the gas heating in Equation (2a), which is in turn much faster than the time scale of the brick cooling in Equation (2b). The algorithm converges rapidly to a solution, which is necessary because the control system must compute a fuel gas flow rate history for the upcoming heating cycle during the five minute pause between cooling and heating cycles. The method is also parameter-free which is needed because the operators have no experience with non-linear differential equation solvers.

References

- Bhatti, M., & Shah, R. (1987). Turbulent and Transition Flow Convective Heat Transfer in Ducts. In Kakac, S., Shah, R., & Aung, W. (Eds.), *Handbook of Single-Phase Convective Heat Transfer*. John Wiley & Sons, Inc., New York, NY.
- Bird, R., Stewart, W., & Lightfoot, E. (1960). *Transport Phenomena*. John Wiley & Sons, Inc., New York, NY.
- Butterfield, P., Schofield, J., & Young, P. (1963). Hot Blast Stoves: Part II. *Journal of the Iron and Steel Institute*, 201(6), 497–508.
- Dembo, R., Eisenstat, S., & Steihaug, T. (1982). Inexact Newton Methods. *SIAM Journal of Numerical Analysis*, 19(2), 400–408.
- Dennis, J., & Schnabel, R. (1983). *Numerical Methods for Unconstrained Optimization and Non-linear Equations*. Computational Mathematics Series. Prentice-Hall, Inc., Englewood Cliffs, NJ.
- Fletcher, R. (1987). *Practical Methods of Optimization* (2nd edition). John Wiley & Sons, Ltd., Chichester, United Kingdom.
- Fox, R., & McDonald, A. (1985). *Fluid Mechanics* (3rd edition). John Wiley & Sons, Inc., New York, NY.

- Hausen, H., & Binder, J. (1962). Vereinfachte Berechnung der Wärmeübertragung durch Strahlung von Einem Gas an Eine Wand. *International Journal of Heat and Mass Transfer*, 5(2), 317–327.
- Hilsenrath, J. (1955). *Tables of Thermal Properties of Gases*. No. 564 in National Bureau of Standards Circular. U.S. Government Printing Office, Washington, DC.
- Hirsch, C. (1988). *Numerical Computation of Internal and External Flows, Volume 1: Fundamentals of Numerical Discretization*. Numerical Methods in Engineering Series. John Wiley & Sons, Ltd., Chichester, United Kingdom.
- Hottel, H. (1954). Radiant Heat Transmission. In McAdams, W. (Ed.), *Heat Transmission* (3rd edition)., Chemical Engineering Series, chap. 4. McGraw-Hill, Inc., New York, NY.
- Jain, A. (1976). Accurate Explicit Equation for Friction Factor. *American Society of Civil Engineers: Journal of the Hydraulics Division*, 102(5), 674–677.
- Kelley, C. (1995). *Iterative Methods for Linear and Nonlinear Equations*, Vol. 16 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Kwakernaak, H., Strijbos, C., & Tijssen, P. (1970). Optimal Operation of Blast Furnace Stoves. *Automatica*, 6(1), 33–40.
- Luenberger, D. (1984). *Linear and Nonlinear Programming* (2nd edition). Addison-Wesley Publishing Co., Inc., Reading, MA.
- Muske, K., Howse, J., Hansen, G., Cagliostro, D., & Chaubal, P. (1999). Hot blast stove process model and model-based controller. *Iron and Steel Engineer*, 23, 56–62.
- Reid, R., Prausnitz, J., & Poling, B. (1987). *The Properties of Gases and Liquids* (4th edition). McGraw-Hill, Inc., New York, NY.
- Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*. PWS Publishing Co., Boston, MA.
- Saad, Y., & Schultz, M. (1986). GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal of Scientific and Statistical Computing*, 7(3), 856–869.
- Schofield, J., Butterfield, P., & Young, P. (1961). Hot Blast Stoves. *Journal of the Iron and Steel Institute*, 199(3), 229–240.
- Singham, J. (1962). Tables of Emissivity of Surfaces. *International Journal of Heat and Mass Transfer*, 5(1), 67–76.
- Sparrow, E., & Cess, R. (1978). *Radiation Heat Transfer* (Augmented edition). Thermal and Fluids Engineering Series. Hemisphere Publishing Corp., Washington, DC.
- Tien, C. (1968). Thermal Radiation Properties of Gases. In Irvine, T., & Hartnett, J. (Eds.), *Advances in Heat Transfer*, Vol. 5. Academic Press, Inc., New York, NY.
- van der Vorst, H., & Vuik, C. (1993). The Superlinear Convergence Behavior of GMRES. *Journal of Computational and Applied Mathematics*, 48(3), 327–341.
- Willmott, A. (1968). Simulation of a Thermal Regenerator under Conditions of Variable Mass Flow. *International Journal of Heat and Mass Transfer*, 11(7), 1105–1116.

A Algebraic Equations of State

In this appendix the algebraic equations of state in Section 2 are discussed. Assuming that the gas is ideal gives the gas density equation

$$\rho_g = \frac{P_g M_g}{\mathcal{R} T_g}, \quad (18)$$

where P_g is the gas pressure, T_g is the gas temperature, M_g is the average molecular weight, and \mathcal{R} is the ideal gas constant.

Assuming that the pressure drop in the stove is due to frictional losses from the gas moving through a pipe results in the relationship

$$\frac{\Delta P_g}{\Delta z} = \frac{\mathcal{F} \dot{m}_g^2}{2 \rho_g \mathcal{D}_g \mathcal{A}_g^2}, \quad (19a)$$

$$\mathcal{F}_w = \frac{1}{\left(3.48 - 1.73 \ln \left(\frac{2\nu}{\mathcal{D}_g} + \frac{42.5}{\text{Re}^{0.9}} \right) \right)^2}, \quad (19b)$$

$$\mathcal{F}_c = \frac{\kappa \mathcal{D}_g}{\mathcal{H}_c}, \quad (19c)$$

$$\mathcal{F} = \mathcal{F}_w + \mathcal{F}_c, \quad (19d)$$

where \mathcal{F} is the friction factor for gas flow in the tubes, and \mathcal{D}_g is the hydraulic diameter of a single tube. Equation (19a) is a correlation discussed in Bird *et al.* (1960) for the pressure drop in a pipe due to friction under turbulent flow conditions. Equation (19b) is a friction factor correlation for rough pipes due to Jain (1976) in which the equivalent sand roughness ν of the checker material is taken to be similar to that of concrete. A value of $\nu = 0.02$ was determined from a plot of equivalent sand roughness for commercial pipe surfaces in Bhatti and Shah (1987). Equation (19c) is an equivalent friction factor derived from the model in Fox and McDonald (1985) for pressure changes due to expansion or contraction in a pipe. Depending on the direction of gas flow, there are either expansions or contractions in the gas tubes due to a slight taper in the tubes of each checker. Specifically, the top diameter of the gas channel is 4 mm less than the bottom diameter. This design is intended to increase the heat transfer in the stove by making the flow more turbulent. The expansion loss coefficient for the heating cycle was chosen as $\kappa = 0.07$, a contraction loss coefficient of $\kappa = 0.06$ was chosen for the cooling cycle, and the average checker length was specified as $\mathcal{H}_c = 15.3 \text{ cm}$. The loss coefficient values were determined from the area ratio between the gas channels at the top and bottom of the checkers as computed in Fox and McDonald (1985).

The heat transfer coefficient h consists of convective and radiative components that are

$$h_c = \frac{k_g}{\mathcal{D}_g} \frac{\left(\frac{\mathcal{F}}{2} \right) (\text{Re} - 1000) \text{Pr}}{1 + \sqrt{\frac{\mathcal{F}}{2}} \left(\left(\sqrt{\sqrt{\frac{\mathcal{F}}{2}}} \nu \text{Re} \right) \left(17.42 - 13.77 (1.01 - 0.99 \text{Pr}^{0.36})^{0.8} \right) - 8.48 \right)}, \quad (20a)$$

$$h_r = \sigma \epsilon_s \frac{(\alpha_h + \alpha_c) T_s^4 - (\epsilon_h + \epsilon_c) T_g^4}{T_s - T_g}. \quad (20b)$$

Equation (20a) is a convective heat transfer correlation for rough pipes described by Bhatti and Shah (1987). In this equation k_g is the gas thermal conductivity, Re is the Reynolds number, and Pr is the Prandtl number. This correlation is valid for Reynolds numbers greater than 2300 and Prandtl numbers greater than 0.5. The Reynolds number range is 2500–5500 during the heating cycle, and 4500–9500 during the cooling cycle. The Prandtl number range is 0.6–0.8 during typical stove operation. Equation (20b) is based on the analysis in Hottel (1954). In this equation σ is the Stephan-Boltzmann constant, ϵ_s is the emissivity of the tube wall, ϵ_h is the water vapor emissivity at T_g , ϵ_c is the carbon dioxide emissivity at T_g , α_h is the water vapor absorptivity at T_s , and α_c is the carbon dioxide absorptivity at T_s . The emissivity and absorptivity values as functions of temperature and pressure for water vapor and carbon dioxide are determined by interpolating the chart data presented in Hottel (1954). For the operating range of the stove, the chart values were determined from experimental measurements of total emission, as reported in Tien (1968) and Sparrow and Cess (1978). The tube wall emissivity ϵ_s is assumed to be that of refractory brick, and is estimated to be 0.8 for each of the checker materials based on data from Singham (1962) and Sparrow and Cess (1978).

The heat capacity C_g , viscosity μ_g , and thermal conductivity k_g as functions of the gas temperature T_g for the gas components during the heating and cooling cycles were determined by fitting the data in Hilsenrath (1955). The resulting interpolating functions are shown in Table 1. When

Gas	$C_g(T_g)$ (cal/gm-K)	$\mu_g(T_g)$ (cm/gm-sec)	$k_g(T_g)$ (cal/cm-sec-K)
CO ₂	$0.3628 - \frac{77.0}{T_g} + \frac{8623.0}{T_g^2}$	$-1.225e-4 + 3.107e-8 T_g + \frac{1.518e-5 \sqrt{T_g}}{\sqrt{T_g}}$	$3.749e-5 + 3.559e-7 T_g - \frac{6.050e-6 \sqrt{T_g}}{\sqrt{T_g}}$
H ₂ O	$1.267 - \frac{34.65}{\sqrt{T_g}} + \frac{378.4}{T_g}$	$-7.962e-4 + 2.963e-5 \sqrt{T_g} + \frac{6.767e-3}{\sqrt{T_g}}$	$-3.977e-4 + 1.514e-5 \sqrt{T_g} + \frac{3.123e-3}{\sqrt{T_g}}$
N ₂	$0.2164 + 7.299e-5 T_g - \frac{947.2}{T_g^2}$	$-1.044e-4 - 2.656e-8 T_g + \frac{1.679e-5 \sqrt{T_g}}{\sqrt{T_g}}$	$-5.330e-5 - 8.804e-9 T_g + \frac{6.842e-6 \sqrt{T_g}}{\sqrt{T_g}}$
O ₂	$0.3003 - \frac{46.64}{T_g} + \frac{6541.0}{T_g^2}$	$-8.908e-5 + 5.972e-8 T_g + \frac{1.602e-5 \sqrt{T_g}}{\sqrt{T_g}}$	$-3.683e-5 + 5.559e-8 T_g + \frac{4.863e-6 \sqrt{T_g}}{\sqrt{T_g}}$
Air	$0.2566 - 1.211e-4 T_g + \frac{2.778e-7 T_g^2 - 1.765e-10 T_g^3 + 3.773e-14 T_g^4}{\sqrt{T_g}}$	$-5.769e-5 + 1.534e-5 \sqrt{T_g} - \frac{4.031e-4}{\sqrt{T_g}}$	$-8.278e-5 + 7.445e-6 \sqrt{T_g} + \frac{2.882e-4}{\sqrt{T_g}}$

Table 1: Interpolating functions for heat capacity C_g , viscosity μ_g , and thermal conductivity k_g with respect to temperature T_g for all the gases needed in the model calculations.

computing the viscosity and thermal conductivity for the blast air during the cooling cycle, the oxygen and moisture injected into this air are ignored because they are each less than 2 mol%. Using the technique in Reid, Prausnitz, and Poling (1987), the pressure correction from 1 atm to the blast pressure of approximately 5 atm is estimated to be less than 1% for the blast air viscosity and less than 4% for the blast air thermal conductivity, and so this effect is also ignored. The viscosity of the waste gas during the heating cycle is determined from the pure component viscosities using the method of Wilke from Reid *et al.* (1987)

$$\mu_m = \sum_{i=1}^n \frac{y_i \mu_i}{\sum_{j=1}^n y_j \phi_{i,j}}, \quad (21)$$

in which μ_m is the viscosity of the mixture, y_i is the mole fraction of component i , μ_i is the viscosity of pure component i , and the interaction parameter $\phi_{i,j}$ is computed by

$$\phi_{i,j} = \frac{\left(1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right)^2}{\sqrt{8 \left(1 + \frac{M_i}{M_j}\right)}}, \quad (22)$$

with M_i the molecular weight of component i . The thermal conductivity of the waste gas is determined from the pure component thermal conductivities using the method of Mason and Saxena in Reid *et al.* (1987)

$$k_m = \sum_{i=1}^n \frac{y_i k_i}{\sum_{j=1}^n 0.85 y_j \phi_{i,j}}, \quad (23)$$

in which k_m is the mixture thermal conductivity, y_i is the mole fraction of component i , k_i is the thermal conductivity of pure component i , and $\phi_{i,j}$ is the interaction parameter shown from Equation (22).